



---

*Research article*

## **Support vector machine optimization via an improved elephant herding algorithm for motor energy efficiency rating**

**Xinrui Ren<sup>1</sup>, Jianbo Yu<sup>2,\*</sup> and Zhaomin Lv<sup>3</sup>**

<sup>1</sup> School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

<sup>2</sup> State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 200433, China

<sup>3</sup> School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai 201620, China

\* **Correspondence:** Email: [jb\\_yu@fudan.edu.cn](mailto:jb_yu@fudan.edu.cn); Tel: +8618818278399.

**Abstract:** Accurate evaluation of motor energy efficiency under off-condition operation can provide an important basis for an energy-saving upgrade of the motor and the elimination of backward motors. By considering the power quality, motor characteristics and load characteristics, a motor energy efficiency evaluation system with seven indexes and 10 grades was constructed. An improved elephant herding optimization method combined with a support vector machine rating model is proposed, it achieved an accuracy higher than 98%. Considering the slow convergence speed and low convergence precision of the standard elephant herding optimization (EHO) method, it is easy to fall into the local optimum problem. To improve population initialization, chaotic mapping and adversarial learning were used to achieve EHO with population diversity and global search capability. Group learning and elite retention have been added to improve the local development ability of the algorithm. The improved EHO has been compared with other intelligent optimization algorithms by using 12 benchmark functions, and the results show that the improved algorithm has better optimization performance.

**Keywords:** motor energy efficiency grading; improved elephant herding optimization; support vector machine

---

### **1. Introduction**

Statistics from the World Energy Agency show that the energy consumption of the industrial sector accounts for nearly 1/3 rd of the global primary energy supply and produces approximately 36% of energy-related CO<sub>2</sub>, and that the electricity consumption of the motor system accounts for more than

50% of the total electricity consumption. Therefore, improving the energy efficiency of the motor system is of great significance for reducing energy consumption, improving cost efficiency, reducing pollution emission and responding to demands for national energy conservation and emission reduction. Cui et al. proposed that a correct rating of the motor energy efficiency is the best way to improve the energy efficiency of motor systems [1].

In recent years, there have been many advances in artificial intelligence technology in various fields [2–6], whereas there are few types of research on the energy efficiency classification of electric motors. In 2012, guo biao (GB) specifies the minimum energy efficiency values for small and medium-sized motors at rated efficiency and provides initial motor energy efficiency standards [7]. However, in the actual production process, the motor is always operating under variable working conditions due to the power quality, load characteristics, and other influential factors. So the motor energy efficiency standard reported in [8] cannot be effectively applied to an actual situation. Luo et al. conducted a comprehensive analysis of the factors affecting the motor energy efficiency and developed a motor energy efficiency evaluation index system [8]. Li has performed an in-depth study of the main factors affecting motor efficiency and used analytic hierarchy process (AHP) to obtain the weight of each index [9]. These studies provide ideas for research on motor energy efficiency rating.

In this paper, the three aspects of power quality, load characteristics and the characteristics of the motor itself are used as the first-level evaluation indicators, also, there are seven second-level indicators that have a greater impact on the energy efficiency of the motor that have been selected and divided into 10-grade intervals. For electrical energy efficiency classes, Ma and Lv used a support vector machine model for classification and achieved an accuracy of 94% with a small number of test samples [10]. The accuracy rate decreased with increasing sample data.

Currently, SVM theory has unique advantages in terms of solving classification and target recognition problems [11], but the key parameters in SVM i.e., the penalty factor  $c$  and kernel parameter  $\sigma$ , are determined by empirical assignment or cross-validation, which are not only inefficient but they are also susceptible to the local optima problem. Metaheuristic algorithm are widely used for the optimization of various complex problems due to their non-derivative mechanism, flexibility, simplicity and ease of implementation. Classical examples of metaheuristics include particle swarm optimization, ant colony optimization, evolutionary algorithms, and simulated annealing. Inspired by more natural biological behavior, the development of this field has recently experienced something similar to a gold rush in terms of a proliferation of similar new or improved algorithms of all kinds. The more recent metaheuristic algorithms are monarch butterfly optimization [12], the whale optimization [13], satin bowerbird optimization [14] the moth search algorithm [15], hunger games search [16], harris hawks optimization [17], the slime mould algorithm [18], runge kutta method [19], the colony predation algorithm [20] the weighted mean of vectors algorithm [21].

The optimization search process for these algorithms has a common feature of using both exploration and development. The exploration phase should utilize and enhance the stochastic operator capability as much as possible to diversify the search in the feature space; the development phase usually follows the exploration phase and should enhance the local search in the vicinity of the more optimal solution. A good optimization algorithm should be able to make a reasonable and fine balance between exploration and exploitation. Otherwise, it is prone to falling into local optimality and premature maturity. Most metaheuristic algorithms have the disadvantage of being insensitive to

the tuning of user-defined parameters and not always converging to the global optimum.

In 2015, wang et al. proposed elephant herd optimization (EHO) inspired by the nomadic behavior of elephant herds [22]. Elephant herds have two special behaviors: 1) the herd is composed of multiple clans and led by the female, and 2) adult males will leave the herd to live alone. Therefore two mechanisms have been developed to mimic the nomadic behavior of elephant herds their, clan renewal mechanism and their separation mechanism. EHO has been shown to be effective at finding optimal solutions [23]. EHO differs from other metaheuristic algorithms in that EHO can generate multiple clans, each of which is a group, and it is a multiple group collaboration mechanism. However, the original EHO only addresses the position update of one clan and does not consider the collaboration of the whole elephant group. The performance of EHO may be significantly improved if the multiple group collaboration approach can be fully utilized and applied.

The accurate evaluation of motor energy efficiency under off-condition operation can provide an important basis for the energy-saving upgrade of the motor and the elimination of backward motors. In the past, research on motor energy efficiency mainly focused on the rated operating condition, and there were few studies on motor energy efficiency under off-conditions.

1) In this article, we present seven evaluation indexes that have a great influence on motor energy efficiency from the three aspects of power quality, load characteristics, as well as motor characteristics, and establish a quantitative classification standard to achieve an accurate grade of the motor energy efficiency.

2) Moreover, we developed an improved EHO (IEHO) algorithm to optimize the motor energy efficiency rating model of SVMs.

3) In order to improve the superior seeking ability of the EHO algorithms, we propose five improvements and compare them with other algorithms.

The organization of this paper is as follows: Section 2 introduces the principle and the optimization search of the standard EHO, and Section 3 introduces the improved EHO, which first analyzes the shortcomings of the standard EHO; we then propose five improvement strategies. Section 4 presents a comparison of the IEHO with other optimization algorithms which was achieved via testing and experimental analysis. Section 5 introduces the model of the energy efficiency evaluation system of electric machines based on the IEHO combined with SVM (IEHO-SVM) algorithm, and experiments and analyses are discussed. The conclusion and future works are discussed in Section 6.

## 2. Elephant herding optimization

The EHO algorithm was inspired by the nomadic behavior of the elephant herd [22]. Elephants live in groups and consist of multiple clans under the leadership of one of the best female elephant ministers. Each clan will select an excellent female elephant to serve as the matriarch, and the matriarch will lead the female elephants or baby elephants who have a direct or indirect blood relationship with her. Male elephants will leave the group to live alone once they become adults. Although male elephants are far away from their family, they can keep in touch with the elephants in the family through low-frequency vibrations.

To make the elephant herding behavior solve global optimization problems, we have established the following idealized rules: 1) An elephant tribe consists of multiple clans and an optimal female elephant female minister, and each clan has a fixed number of elephants. 2) Each clan in each

generation elects a superior matriarch, and then a fixed number of male elephants leave, followed by an equal number of new ones.

### 2.1. Clan updating operator

The position of each elephant in the clan is affected by the matriarch. For the elephant  $j$  in the clan  $ci$ , the position is updated as follows:

$$x_{ci,j}^{new} = x_{ci,j} + \alpha (x_{best,ci} - x_{ci,j}) r \quad (2.1)$$

where  $\alpha \in [0, 1]$  represents the scaling factor,  $x_{ci,j}^{new}$  represents the new position of Elephant  $j$  in the clan  $ci$ ,  $x_{ci,j}$  represents the old position of Elephant  $j$  in the clan  $ci$ ,  $x_{best,ci}$  represents the matriarch in the clan  $ci$  and  $r \in [0, 1]$  is a random number.

The position of the matriarch has a specific update formula, and the updated position is as follows:

$$x_{best,ci}^{new} = \beta \times x_{center,ci} \quad (2.2)$$

where  $\beta \in [0, 1]$  represents a control parameter that determines the factor of  $x_{center,ci}$  influence on  $x_{best,ci}$ .  $x_{center,ci}$  represents the center position of the clan  $ci$ . It can be determined by the following equation.

$$x_{center,ci,d} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j,d} \quad (2.3)$$

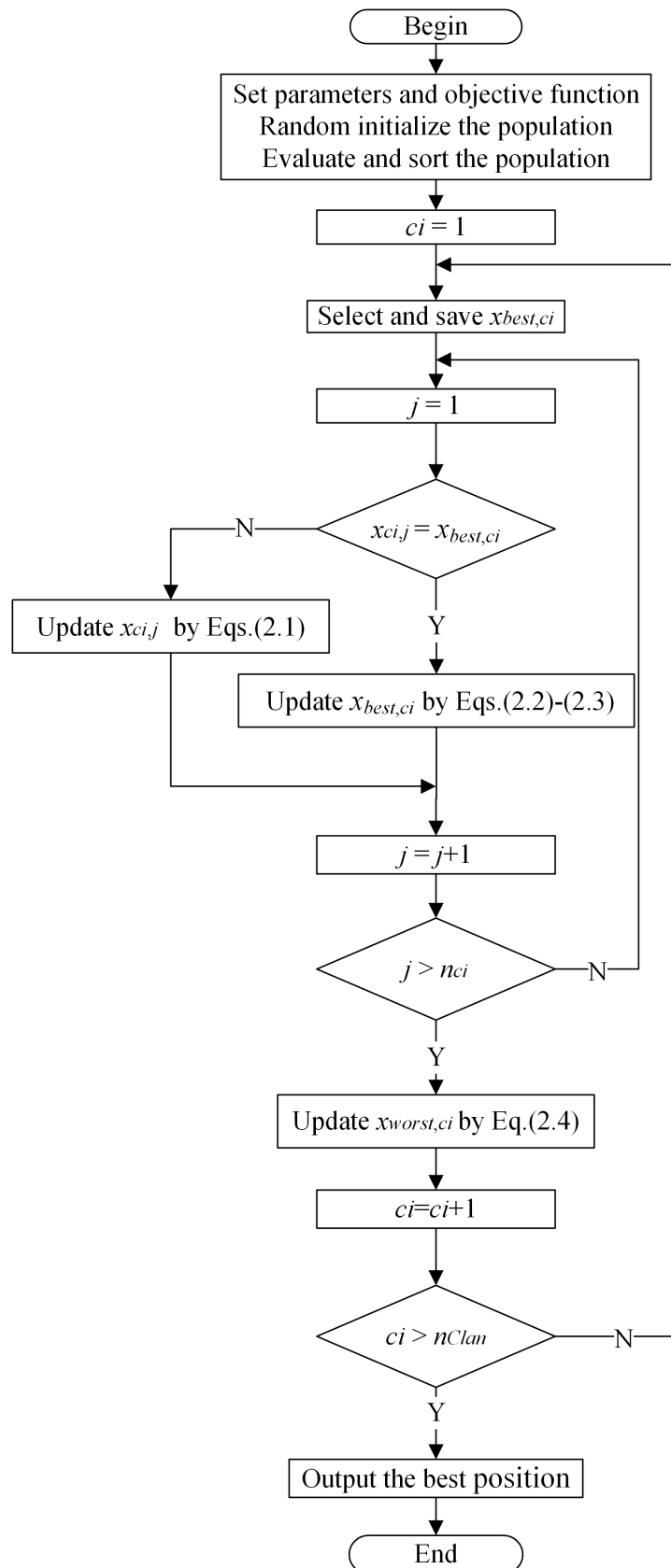
where  $d \in [1, D]$  represents the  $d$  th dimension,  $D$  is the total number of dimensions and  $n_{ci}$  represents the number of elephants of Clan  $ci$ .

### 2.2. Separating operator

In the elephant group, the male elephant in adulthood prefers to leave the group to live alone. To simulate this process, the adult elephant is regarded as the least adaptive elephant in the algorithm. At the same time, to make the algorithm have greater searchability, it is assumed that the elephant with the worst fitness work the separation operator shown in Eq 2.4 in each generation:

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times r \quad (2.4)$$

where  $x_{max}$  and  $x_{min}$  respectively represent the maximum and minimum values of the search space,  $x_{worst,ci}$  represents the position of the worst elephant in the clan  $ci$ . The EHO algorithm flowchart is shown in Figure 1.



**Figure 1.** EHO algorithm flowchart.

### 3. Improved Elephant herding optimization algorithm

The standard EHO algorithm updates the position of each elephant by using clan update and separation operations. Its position update method is singular and has the following disadvantages:

1) The initial position of the elephant is randomly generated, which affects the efficiency of elephant swarm optimization.

2) It only utilizes the position variable and ignores the fact that the moving speed will affect the convergence speed of the elephant group.

3) In the clan update process, the position update of the best female elephant in the entire elephant group is not considered. For ordinary elephants, the position is updated according to the position of themselves and that of the matriarch. The matriarch position is updated by the middle position of the clan, making it easy to form the local optimal position.

4) In the iteration process, the optimal solution of the clan is updated and not retained. Equation 2.4 was used in the separation process to replace the worst elephant in each generation of the clan, simulating the adult male elephant leaving and producing a new baby elephant. Although the ethnic diversity was preserved, as a newborn baby elephant, it would be protected by other elephants and should have a good position.

Based on these shortcomings, we propose the following improvement strategies.

#### 3.1. Strategy 1: Chaotic mapping and opposition-based learning population initialization

The distribution and initial value of the initial position of the population affect the optimization efficiency of the population. It is evenly distributed in the solution space, which is beneficial to improve the population diversity and global searching ability.

This paper introduces a population initialization strategy based on chaotic mapping [24] and opposition-based learning [25]. The strategy initializes the population by using better randomness, spatial ergodicity and non-repetition of chaotic mapping; it also uses the opposition-based learning method to generate its corresponding opposite individuals. Then, it compares and selects individuals with good fitness values as the initial individuals to improve the overall performance of the algorithm optimization efficiency.

Chaotic mapping is used to generate chaotic sequences and random sequences are generated by simple deterministic systems. Table 1 shows several chaotic mapping functions in common use.

**Table 1.** Common chaotic mapping functions.

Chaotic mapping function	Expression
Logistic	$Z^{k+1} = \lambda Z^k (1 - Z^k), 0 < \lambda < 4$
ICMIC	$Z^{k+1} = \sin(\alpha/Z^k), \alpha \in (0, \infty)$
Chebyshev	$Z^{k+1} = \cos(k \cos^{-1} Z^k), k = 4, Z^k \in (0, 1)$
Tent	$Z^{k+1} = \begin{cases} Z^k/0.7, Z^k \in (0, 0.7] \\ 10/3Z^k(1 - Z^k), Z^k \in (0.7, 1) \end{cases}$

According to the analysis of the study presented in [26], when  $\lambda = 4$ , the distribution of a logistic

mapping chaotic sequence is unevenly distributed, and the optimization speed and efficiency are slow. When the initial value is 0 or is fixed, the ICMIC mapping function cannot continue to iterate. Chebyshev mapping has higher requirements for functions. The chaotic sequence of tent mapping is evenly distributed, the optimization is faster and the search efficiency is higher. Tent mapping has been applied in this study to generate the initial population of a chaotic sequence.

The initialization steps are as follows:

Step 1: Generate random numbers:

$$Z_{ci,j,d}^k \in (0, 1) \quad (3.1)$$

where  $d$  represents the dimension  $d = 1, 2, \dots, D$ ;  $K$  represents the number of current iterations.

Step 2: Calculate the tent function chaotic map:

$$Z_{ci,j,d}^{k+1} = \begin{cases} Z_{ci,j,d}^k/0.7, Z_{ci,j,d}^k \in (0, 0.7] \\ 10/3Z_{ci,j,d}^k(1 - Z_{ci,j,d}^k), Z_{ci,j,d}^k \in (0.7, 1) \end{cases} \quad (3.2)$$

Step 3: Generate  $N$  positions, which gives the original population  $X = (x_{ci,j,1}, x_{ci,j,2}, \dots, x_{ci,j,D})$  and  $N = ci \times j$ :

$$x_{ci,j,d} = x_{min,d} \times (1 - Z_{ci,j,d}^{k+1}) + x_{max,d} Z_{ci,j,d}^{k+1} \quad (3.3)$$

where  $x_{max,d}$  and  $x_{min,d}$  represent the upper and lower limits of elephants in the  $D$  dimension.

Step 4: Perform opposition-based population initialization  $OX = (ox_{ci,j,1}, ox_{ci,j,2}, \dots, ox_{ci,j,D})$  based on opposition-based learning:

$$ox_{ci,j,d} = x_{min,d} Z_{ci,j,d}^{k+1} + x_{max,d} \times (1 - Z_{ci,j,d}^{k+1}) \quad (3.4)$$

where  $ox_{ci,j,d}$  represents the individual opposing  $x_{ci,j,1}$  and  $X$  and  $OX$  are opposing populations.

Step 5: Merge the original population and the opposing population, calculate the individual fitness, and take the first  $N$  positions with high fitness as the initial population.

### 3.2. Strategy 2: Speed

Inspired by the PSO algorithm [27], the individual elephants were given a speed to simulate elephant motion; additionally a global speed strategy is proposed, which can accelerate the convergence speed of the elephant herd and balance the convergence speed by introducing inertia weights to avoid premature population maturity.

Speed is also initialized by chaotic mapping and opposition-based learning, taken as  $v_{max,d} = 0.2(x_{max,d} - x_{min,d})$  and  $v_{max,d} = -v_{min,d}$ . The elephant's position is updated as follows:

$$\begin{aligned} v_{ci,j}^{new} &= \omega^k v_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \\ x_{ci,j}^{new} &= x_{ci,j} + v_{ci,j}^{new} \end{aligned} \quad (3.5)$$

where  $x_{ci,j}^{new}$  and  $v_{ci,j}^{new}$  represent the updating position and speed of Elephant  $j$  in Clan  $ci$  respectively, and  $\omega^k$  represents the linearly decreasing inertia weight in the iterative process.

$$\omega^k = \frac{(\omega^{begin} - \omega^{end})(K_{max} - k)}{K_{max}} + \omega^{end} \quad (3.6)$$

where  $\omega^{begin}$  and  $\omega^{end}$  represent the weight at the beginning and end of the iteration respectively, taken as  $\omega^{begin} = 0.9$  and  $\omega^{end} = 0.4$ ;  $K_{max}$  represents the maximum number of iterations.

### 3.3. Strategy3: Adaptive learning step size

In the above equation,  $a$  represents the step size from the current position of the elephant to the optimal position. When the value of  $a$  is large, the global optimization ability is strong, while the local optimization ability is poor; when the value of  $a$  is small, the global optimization ability is poor, while the local optimization ability is strong. In the standard EHO algorithm, the value of  $a$  is fixed and the superior and inferior individuals in the clan have the same search range, which does not match the actual situation. The actual situation is that the better individual is close to the optimal solution, so a smaller learning step size should be used to enhance the local optimization ability; alternatively, the less fit individual is far away from the optimal solution, so a larger learning step size should be used to enhance the global optimization ability. Therefore, an adaptive learning step size strategy is proposed.

1). The optimal solution  $x_{mbest}$  of the clan requires less learning, learning step take  $a$  fixed value and taken as  $a_1 = 0.2$ .

2). The optimal solution  $x_{cbest,ci}$  of the clan is learned from the optimal solution  $x_{mbest}$  of the clan.

Learning step  $a_{cbest,ci}$  is expressed as follows:

$$a_{cbest,ci} = a_2 \left( \frac{k}{K_{max}} \right) \times \frac{f(x_{cbest,ci})}{f(x_{mbest})} \quad (3.7)$$

where  $f(x_{cbest,ci})$  and  $f(x_{mbest})$  represent the fitness values of  $x_{cbest,ci}$  and  $x_{mbest}$  respectively.

3). Other elephants  $x_{otherci,j}$  of the clan learn from the matriarch  $x_{cbest,ci}$  of the clan. Learning step  $a_{otherci,j}$  is expressed as follows:

$$a_{otherci,j} = a_3 \left( \frac{k}{K_{max}} \right) \times \frac{f(x_{otherci,j})}{f(x_{cbest,ci})} \quad (3.8)$$

where  $f(x_{otherci,j})$  and  $f(x_{cbest,ci})$  represent the fitness values of  $x_{otherci,j}$  and  $x_{cbest,ci}$  respectively and  $a_1 < a_2 < a_3$  are constants.

### 3.4. Strategy 4: Group learning

In a large elephant herd tribe, each individual has a unique level of fitness. There are excellent female ministers and matriarchs and males and other ordinary elephants, so the elephants have been divided into four groups, and different groups of elephants have different learning styles. The first group is the best female elephant of the tribe, represented by  $x_{mbest}$ . The second group is the best matriarch of the clan, represented by  $x_{cbest,ci}$ . The third group is the other female elephants and young elephants, represented by  $x_{otherci,j}$ . The fourth group is the worst individual of the clan, represented by  $x_{worst,ci}$ .

The learning mode of the first group  $x_{mbest}$  is as follows:

As the best female elephant in the herd, her position and speed should be updated according to the matriarch information of all clans. The update mode is as follows:

$$\begin{aligned} v_{mbest}^{new} &= \omega^k \times v_{mbest} + a_1 (x_{center} - x_{mbest}) r \\ x_{mbest}^{new} &= x_{mbest} + v_{mbest}^{new} \\ x_{center} &= \frac{1}{n_{ci}} \times \sum_{i=1}^{n_{ci}} x_{cbest,ci} \end{aligned} \quad (3.9)$$



where  $x_{mbest}^{new}$  and  $v_{mbest}^{new}$  respectively represent the update position and speed of  $x_{mbest}$ ,  $x_{center}$  represents the central location for all matriarchs and  $n_{ci}$  represents the total number of clans.

The learning mode of the second group  $x_{cbest,ci}$  is as follows:

The best matriarch of the clan should learn from the best female elephant of the tribe to achieve a larger search area. The update mode is as follows:

$$\begin{aligned} v_{cbest,ci}^{new} &= \omega^k v_{cbest,ci} + a_{cbest,ci} (x_{mbest} - x_{cbest,ci}) r \\ x_{cbest,ci}^{new} &= x_{cbest,ci} + v_{cbest,ci}^{new} \end{aligned} \quad (3.10)$$

where  $x_{cbest,ci}^{new}$  and  $v_{cbest,ci}^{new}$  respectively represent the update position and speed of  $x_{cbest,ci}$ .

The learning mode of the third group  $x_{otherci,j}$  is as follows:

The other female elephants and young elephants should learn from the best matriarch of the clan to achieve a faster learning speed. The update mode is as follows:

$$\begin{aligned} v_{otherci,j}^{new} &= \omega^k v_{otherci,j} + a_{otherci,j} (x_{cbest,ci} - x_{otherci,j}) r \\ x_{otherci,j}^{new} &= x_{otherci,j} + v_{otherci,j}^{new} \end{aligned} \quad (3.11)$$

where  $x_{otherci,j}^{new}$  and  $v_{otherci,j}^{new}$  respectively represent the update position and speed of  $x_{cbest,ci}$ .

In order to optimize the learning of the elephants in the clan, a mutual learning strategy is proposed. The ordinary elephants in the clan learn from the matriarch first, and then from others who are better than themselves.

Elephant  $p$  is randomly selected from Clan  $ci$ ; then, its new fitness value is calculated. If the fitness value of Elephant  $j$  is better than that of Elephant  $p$ , Elephant  $p$  needs to learn from Elephant  $j$ . If it is the opposite Elephant  $j$  needs to learn from Elephant  $p$ . If the fitness values of Elephants  $j$  and  $p$  are equal, then there is no need to learn from each other. The update mode is as follows:

$$\begin{cases} f(x_{otherci,j}^{new}) < f(x_{otherci,p}^{new}), \hat{v}_{otherci,j}^{new} = v_{otherci,j}^{new} + r(x_{otherci,p}^{new} - x_{otherci,j}^{new}) \\ f(x_{otherci,j}^{new}) > f(x_{otherci,p}^{new}), \hat{v}_{otherci,j}^{new} = v_{otherci,j}^{new} + r(x_{otherci,p}^{new} - x_{otherci,j}^{new}) \\ \hat{x}_{otherci,j}^{new} = x_{otherci,j}^{new} + \hat{v}_{otherci,j}^{new} \end{cases} \quad (3.12)$$

where  $x_{otherci,j}^{new}$  and  $v_{otherci,j}^{new}$  respectively represent the position and speed of Elephant  $p$  after learning.

### 3.5. Strategy 5: Elite retention

In the standard EHO algorithm, the separation operator is used to replace the worst solution in the clan. At the same time, the clan optimal solution continuously updated during the iteration, and not preserved. Therefore, an elite retention strategy was developed. The worst elephants in the fourth group directly learned from the matriarch and replaced the worst solution with the clan optimal solution. At the same time, the fitness values for each elephant before and after updating were compared, and the positions of elephants with high fitness values were retained. The method preserves the optimal solution for each generation's clan and the optimal solution for individual elephants, which can effectively accelerate the optimization speed of the elephant group.

The IEHO algorithm flowchart is shown in Figure 2.

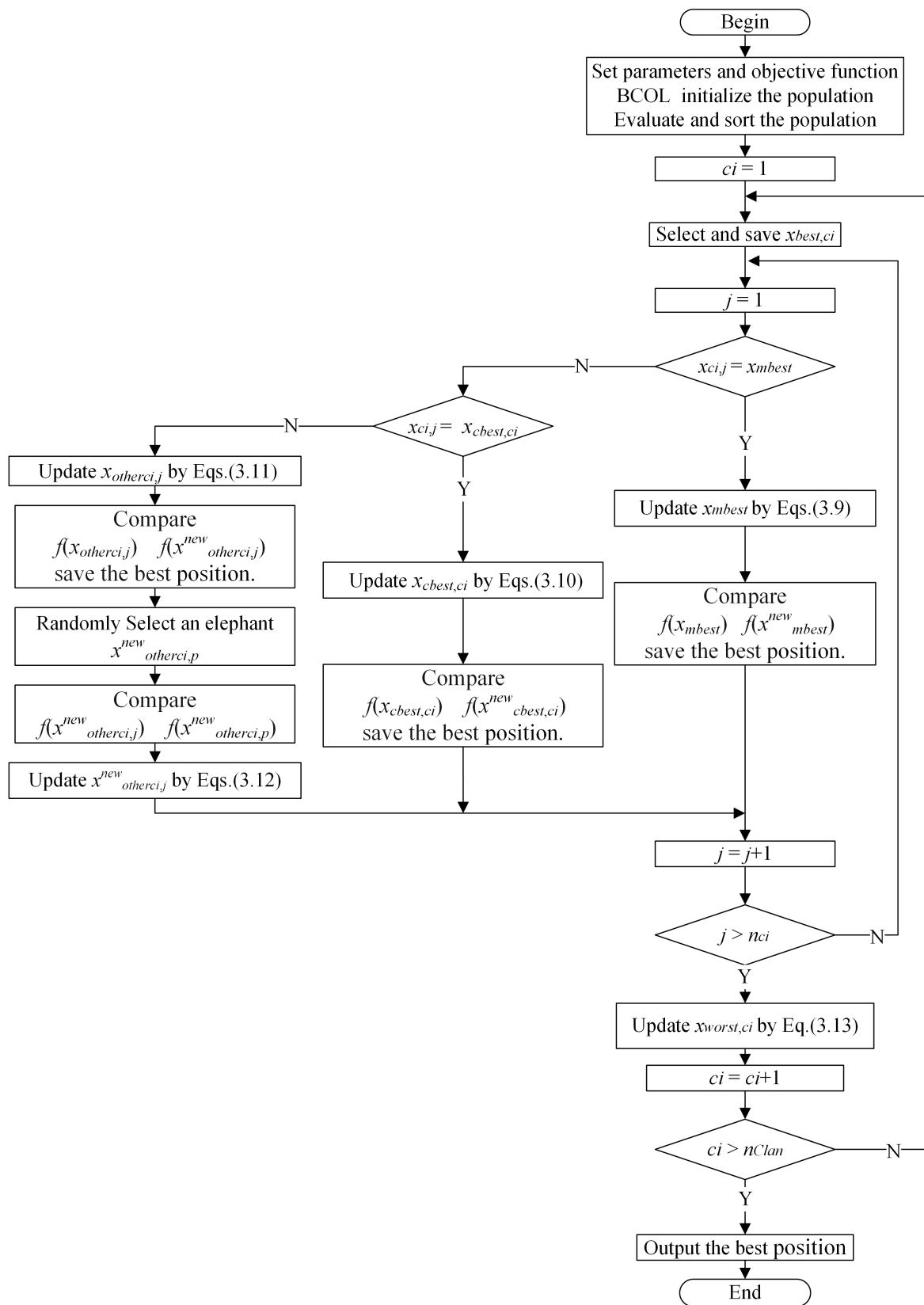


Figure 2. IEHO algorithm flowchart.

The learning mode of the fourth group  $x_{worst,ci}$  is as follows: the worst male elephant learns directly from the matriarch. The update mode is as follows:

$$x_{worst,ci}^{new} = x_{cbest,ci} \quad (3.13)$$

### 3.6. Computational complexity analysis

Standard EHO mainly includes initialization, fitness evaluation, sorting and optimal selection, a clan updating operator and a separating operator. In the associated formula,  $M$  indicates the number of clans,  $N$  indicates the number of elephants in each clan,  $D$  is the number of dimensions of the problem and  $T$  represents the maximum number of iterations. The complexity of the initialization stage is  $O(1)$ . In the main part of the algorithm, the complexity of the sorting algorithm is  $O(N * M \log(N * M))$ . The clan updating operator is  $O(M * N)$ ; the separating operator is  $O(N)$ . The overall complexity of the algorithm is  $O(1 + T * (N * M \log N * M + M * N + M))$ .

The proposed IEHO algorithm mainly includes the following parts: initialization, fitness evaluation, sorting and optimal selection, position and speed updating and a fitness comparison. During the initial stage, the complexity of initializing the individuals in each dimension based on dyadic learning is  $O(M * N * D)$ . In the main part of the algorithm, the complexity of the sorting algorithm is  $O(N * M \log(N * M))$ , the complexity of updating the speed and position of the individuals is  $O(M * N)$  and the complexity of adaptive comparison is  $O(M * N)$ . From the above analysis, we can acquire the complexity of the whole algorithm:  $O(M * N * (D + T * (\log(M * N) + 2)))$ .

## 4. Testing and experimental analysis

### 4.1. Experiment design

To verify the effectiveness of the IEHO algorithm, 12 typical reference functions proposed in [28] were selected for testing, as is shown in Table 2.  $f_1 - f_6$  are unimodal functions with only one global optimal point, which is used to test the convergence accuracy of the algorithm.  $f_7 - f_{12}$  are multi-peak functions with multiple local optimal points, which are used to test the algorithm's global search capability, local development and ability to jump out of local optima.

Due to the fact that five improved strategies are proposed, to prove the effectiveness of the final five strategy combinations of IEHO, we did not consider partial strategies based on the EHO algorithm that are stronger than the IEHO algorithm, and a strategy combination comparison experiment was conducted. First, a comparison experiment of strategy combinations was conducted, and *Schweffel2.22* and *Qing* were selected as the benchmark test functions; the experimental results are shown in Figure 3. EHO + S1 is the combination of EHO with Strategy 1; EHO + S12 is the combination of EHO with Strategy 1 and Strategy 2; EHO + S123 is the combination of EHO with Strategy 1, Strategy 2 and Strategy 3; EHO + S1234 is the combination of EHO with Strategy 1, Strategy 2, Strategy 3 and Strategy 4; EHO + S12345 is the combination of EHO with 5 strategies. Furthermore, to prove the convergence speed, convergence accuracy and stability of IEHO, the IEHO algorithm has been compared with the multi-mechanism hybrid EHO (MCEHO) algorithm proposed in [21], original EHO algorithm [22], the PSO algorithm [29] and the artificial bee colony (ABC) algorithm [30, 31].

To ensure fairness and accuracy of the algorithm comparison, the parameters of the five algorithms were set to be the same, the population size was set as 50 and the maximum number of iterations was set to be 500. For EHO, MCEHO and IEHO, we set  $nclan = 5$  and  $nci = 10$ . Each algorithm ran the optimal value independently 30 times on each test function in 30D and 100D. The mean value and standard deviation were calculated to evaluate the convergence accuracy and stability of each algorithm. The simulation environment was a Windows 10 operating system equipped with an i5-6500 CPU, 3.2 GHz main frequency, 8 GB of memory, and MATLAB 2016a simulation software.

**Table 2.** Benchmark test functions.

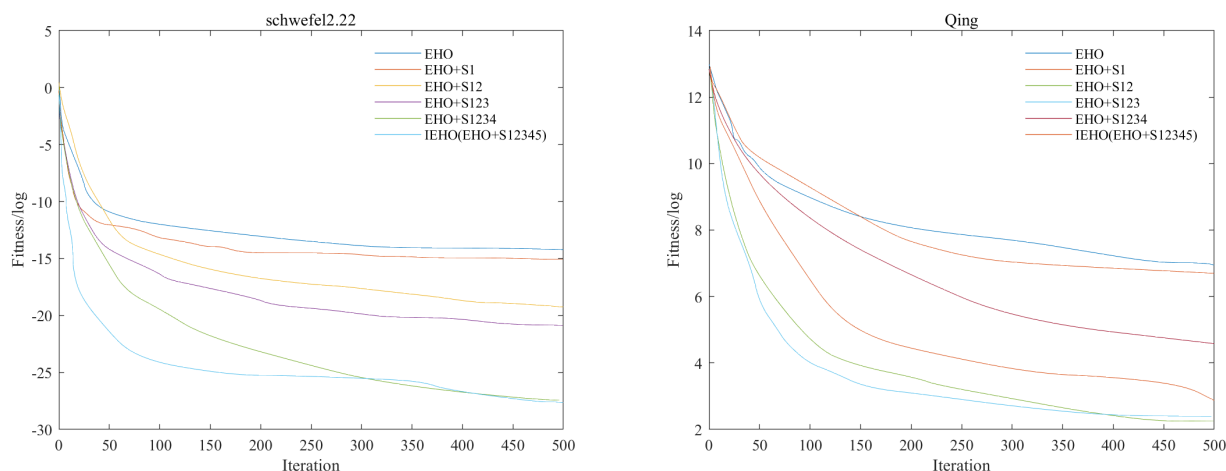
Function	Expression	Search range	Extremum
Schwefel2.22	$f_1(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	[-10.00, 10.00]	0
Schwefel2.21	$f_2(x) = \max \{ x_i , 1 \leq i \leq n\}$	[-100.00, 100.00]	0
Sum of different powers	$f_3(x) = \sum_{i=1}^n  x_i ^{i+1}$	[-1.00, 1.00]	0
Zakharov	$f_4(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	[-5.00, 10.00]	0
Sphere	$f_5(x) = \sum_{i=1}^n x_i^2$	[-100.00, 100.00]	0
Qing	$f_6(x) = \sum_{i=1}^n (x_i^2 - i)^2$	[-500.00, 500.00]	0
Griewank	$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-300.00, 300.00]	0
Rastrigin	$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	0
Levy	$f_9(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_n - 1)^2 [1 + \sin^2(2\pi w_n)]$	[-10.00., 10.00]	0
Schaffers F7	$f_{10}(x) = \left[ \frac{1}{n-1} \sum_{i=1}^{n-1} \left( \sqrt{s_i} \left( \sin(50s_i^{0.2}) + 1 \right) \right)^2 \right]^2$ , $s_i = \sqrt{x_i^2 + x_{i+1}^2}$	[-10.00, 10.00]	0
Ackley	$f_{11}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$	[-32.00, 32.00]	0
Salomon	$f_{12}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	[-100.00, 100.00]	0

#### 4.2. Experimental results and analysis

It can be seen in Figure 3 that with the increase of strategy combinations, the EHO performance gradually improved, the speed and adaptive learning step size can effectively improve the convergence speed, group learning and elite retention can effectively improve the convergence accuracy and the IEHO performance from the combination of five strategies was optimal. The IEHO still follows the nomadic behavior of the elephant herd, IEHO is an improvement on the standard EHO, only its search method is improved during the iterative process so that the elephant herd becomes more intelligent; essentially, it is still EHO. These five strategies are also useful for improving the performance of other optimization algorithms.

Table 3 shows the means and standard deviations of the fitness values for the 12 test functions calculated by the five algorithms under the conditions of fixed iteration times for the low dimension (30) and high dimension (100), respectively. The performance of each of the five algorithms on each test function is shown in Figure 6, and the three-dimensional mathematical model of each of the 12 test

functions is also presented.



(a) Convergence curve for Schwefel2.22

(b) Convergence curve for Qing

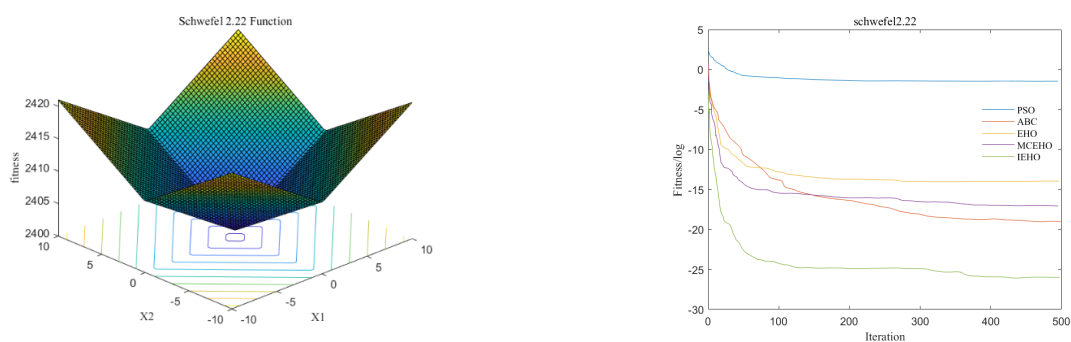
**Figure 3.** Strategy combination comparison test results.

The left plots in Figures 4–6 show the three-dimensional search spatial distributions for the 12 test functions.  $X_1$  and  $X_2$  are function search intervals, and the vertical axis is the function value. The graphs on the right are the convergence curves for the intelligent algorithms for optimizing the test functions. The horizontal axis is the number of iterations, and the vertical axis is the logarithm of the optimal values.

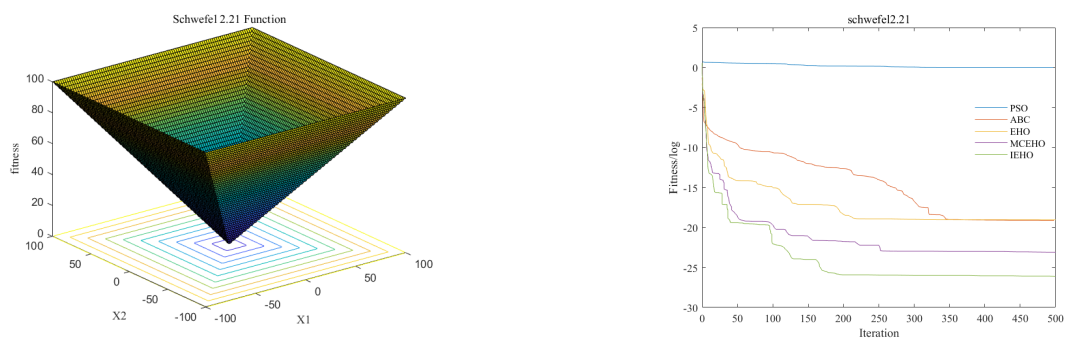
As can be seen in Figures 4–6, after 500 iterations, all five algorithms were able to converge to the extreme value of the test function with different convergence speeds and convergence accuracies; however, PSO resulted in the worst performance. For the unimodal functions  $f_1 - f_6$ , in the beginning, the convergence rates of EHO and IEHO were faster and better than that of MCEHO. This shows that the initial population initialization strategy based on chaotic mapping and opposition-based learning can facilitate a better position and fitness value for the initial population and improve the global searching ability. The convergence speed of the IEHO algorithm improved after 100 iterations, while that of the MCEHO algorithm improved after 150 iterations. The convergence speeds of the EHO and ABC is similar, but slightly higher than ABC. The IEHO algorithm was able to converge to the minimum value after about 100 iterations, and the convergence accuracy was higher than that of other algorithms, indicating that the addition of a speed strategy can effectively accelerate the population optimization speed. Regarding the unimodal function  $f_3$ , the IEHO algorithm was able to converge to 0 and the standard deviation was 0.

Table 3. Comparison results for the five algorithms for the test functions process.

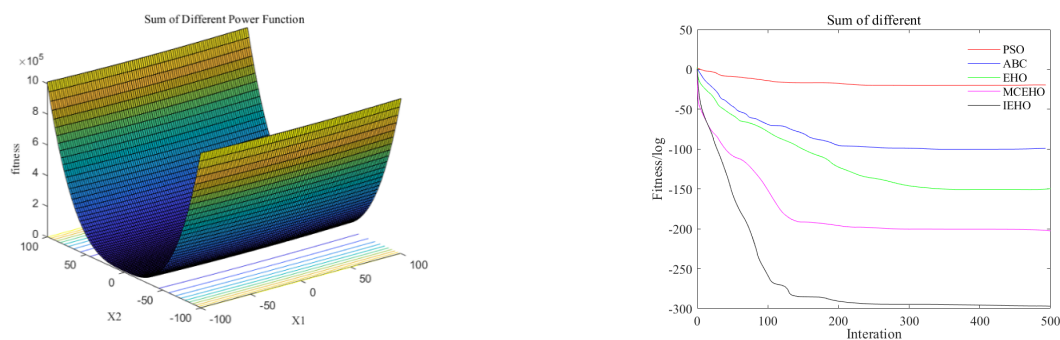
f	D	PSO		ABC		EHO		MCEHO		IEHO	
		Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)	Ave (Std)
$f_1$	30	0.981 (1.13)	$9.04 \times 10^{-19}$ ( $3.98 \times 10^{-19}$ )	$6.38 \times 10^{-14}$ ( $5.18 \times 10^{-14}$ )	$1.48 \times 10^{-17}$ ( $1.42 \times 10^{-17}$ )	$1.61 \times 10^{-26}$ ( $2.23 \text{ times } 10^{-26}$ )					
	100	2.43 (3.45)	$7.85 \times 10^{-18}$ ( $6.56 \times 10^{-18}$ )	$3.89 \times 10^{-13}$ ( $2.18 \times 10^{-13}$ )	$2.76 \times 10^{-16}$ ( $1.43 \times 10^{-16}$ )	$4.91 \times 10^{-25}$ ( $3.29 \times 10^{-25}$ )					
$f_2$	30	6.59 (2.37)	$8.54 \times 10^{-19}$ ( $4.74 \times 10^{-19}$ )	$5.29 \times 10^{-18}$ ( $4.21 \times 10^{-19}$ )	$2.31 \times 10^{-23}$ ( $1.78 \times 10^{-23}$ )	$1.76 \times 10^{-26}$ ( $2.48 \times 10^{-26}$ )					
	100	30.1 (28.9)	$7.31 \times 10^{-18}$ ( $5.98 \times 10^{-18}$ )	$3.88 \times 10^{-17}$ ( $3.29 \times 10^{-17}$ )	$3.78 \times 10^{-22}$ ( $2.91 \times 10^{-22}$ )	$4.79 \times 10^{-25}$ ( $2.77 \times 10^{-25}$ )					
$f_3$	30	$9.13 \times 10^{-12}$ ( $3.02 \times 10^{-11}$ )	0.00 (0.00)	0.0 (0.00)	0.00 (0.00)	0.00 (0.00)					
	100	$8.67 \times 10^{-10}$ ( $7.54 \times 10^{-10}$ )	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)					
$f_4$	30	57.8 (24.7)	$2.09 \times 10^{-27}$ ( $1.15 \times 10^{-27}$ )	$4.11 \times 10^{-30}$ ( $3.78 \times 10^{-30}$ )	$8.85 \times 10^{-36}$ ( $2.86 \times 10^{-36}$ )	$3.21 \times 10^{-47}$ ( $1.76 \times 10^{-47}$ )					
	100	3.78 (1.88)	$4.93 \times 10^{-26}$ ( $2.78 \times 10^{-26}$ )	$2.78 \times 10^{-29}$ ( $1.66 \times 10^{-29}$ )	$4.66 \times 10^{-35}$ ( $4.01 \times 10^{-35}$ )	$3.76 \times 10^{-47}$ ( $2.45 \times 10^{-47}$ )					
$f_5$	30	7.75 ( $1.95 \times 10^{-10}$ )	$2.41 \times 10^{-14}$ ( $2.51 \times 10^{-14}$ )	$3.39 \times 10^{-16}$ ( $2.99 \times 10^{-16}$ )	$5.54 \times 10^{-18}$ ( $1.68 \times 10^{-18}$ )	$2.18 \times 10^{-28}$ ( $4.67 \times 10^{-27}$ )					
	100	9.36 (6.5)	$6.89 \times 10^{-13}$ ( $6.01 \times 10^{-13}$ )	$4.18 \times 10^{-15}$ ( $3.32 \times 10^{-15}$ )	$8.96 \times 10^{-17}$ ( $9.03 \times 10^{-17}$ )	$6.69 \times 10^{-27}$ ( $7.43 \times 10^{-27}$ )					
$f_6$	30	$2.12 \times 10^7$ ( $1.42 \times 10^7$ )	$4.46 \times 10^8$ ( $3.74 \times 10^8$ )	$3.89 \times 10^7$ ( $4.18 \times 10^7$ )	$4.01 \times 10^5$ ( $9.73 \times 10^4$ )	$6.58$ ( $2.14 \times 10^2$ )					
	100	$3.78 \times 10^8$ ( $2.38 \times 10^8$ )	$5.67 \times 10^9$ ( $5.13$ )	$5.88 \times 10^8$ ( $4.99 \times 10^8$ )	$3.09 \times 10^6$ ( $2.98 \times 10^6$ )	$8.37 \times 10^2$ ( $4.76 \times 10^2$ )					
$f_7$	30	$2.43 \times 10^{-4}$ ( $1.03 \times 10^{-4}$ )	$8.63 \times 10^{-6}$ ( $2.54 \times 10^{-6}$ )	$5.17 \times 10^{-8}$ ( $4.11 \times 10^{-8}$ )	$1.96 \times 10^{-10}$ ( $1.03 \times 10^{-10}$ )	$4.58 \times 10^{-16}$ ( $2.17 \times 10^{-20}$ )					
	100	$8.49 \times 10^{-3}$ ( $6.29 \times 10^{-3}$ )	$3.71 \times 10^{-5}$ ( $4.78 \times 10^{-5}$ )	$2.97\text{E-}07$ ( $8.92\text{E-}08$ )	$3.88\text{E-}09$ ( $1.19\text{E-}09$ )	$8.73 \times 10^{-15}$ ( $2.19 \times 10^{-17}$ )					
$f_8$	30	42.1 (15.4)	$3.75 \times 10^{-5}$ ( $9.06 \times 10^{-6}$ )	$3.88 \times 10^{-5}$ ( $2.17 \times 10^{-5}$ )	$3.19 \times 10^{-6}$ ( $7.42 \times 10^{-7}$ )	$6.77 \times 10^{-15}$ ( $5.23 \times 10^{-16}$ )					
	100	49.4 (40.1)	$5.13 \times 10^{-4}$ ( $4.99 \times 10^{-4}$ )	$5.22 \times 10^{-4}$ ( $4.89 \times 10^{-4}$ )	$7.81 \times 10^{-5}$ ( $5.95 \times 10^{-5}$ )	$8.98 \times 10^{-14}$ ( $5.22 \times 10^{-16}$ )					
$f_9$	30	1.72 (0.137)	2.31 (0.79)	0.218 (0.133)	$1.25$ ( $3.1 \times 10^{-2}$ )	$1.62 \times 10^{-2}$ ( $0.79 \times 10^{-2}$ )					
	100	4.09 (0.393)	4.38 (8.21)	8.22 (7.17)	4.81 (3.97)	<b>1.19 (0.482)</b>					
$f_{10}$	30	7.65 (0.798)	$2.21 \times 10^{-9}$ ( $1.23 \times 10^{-10}$ )	$2.19 \times 10^{-10}$ ( $2.18 \times 10^{-10}$ )	$2.23 \times 10^{-12}$ ( $1.13 \times 10^{-12}$ )	$1.84 \times 10^{-20}$ ( $1.07 \times 10^{-20}$ )					
	100	24.3 (3.25)	$3.34 \times 10^{-8}$ ( $5.88 \times 10^{-9}$ )	$5.39 \times 10^{-9}$ ( $4.18 \times 10^{-9}$ )	$4.78 \times 10^{-11}$ ( $3.89 \times 10^{-11}$ )	$1.09 \times 10^{-10}$ ( $0.48 \times 10^{-19}$ )					
$f_{11}$	30	1.93 (0.678)	$8.45 \times 10^{-16}$ ( $8.12 \times 10^{-16}$ )	$2.31 \times 10^{-13}$ ( $5.19 \times 10^{-12}$ )	$3.94 \times 10^{-14}$ ( $3.86 \times 10^{-14}$ )	$8.13 \times 10^{-18}$ ( $8.01 \times 10^{-18}$ )					
	100	2.19 (49.1)	$3.99 \times 10^{-18}$ ( $4.13 \times 10^{-17}$ )	$3.11 \times 10^{-12}$ ( $2.91 \times 10^{-11}$ )	$6.83 \times 10^{-14}$ ( $5.39 \times 10^{-15}$ )	$3.88 \times 10^{-17}$ ( $0.19 \times 10^{-17}$ )					
$f_{12}$	30	0.313 (4.38)	$4.53 \times 10^{-9}$ ( $3.98 \times 10^{-9}$ )	$3.87 \times 10^{-10}$ ( $3.09 \times 10^{-10}$ )	$5.76 \times 10^{-21}$ ( $4.91 \times 10^{-21}$ )	$8.15 \times 10^{-27}$ ( $7.16 \times 10^{-27}$ )					
	100	1.19 (0.159)	$2.17 \times 10^{-8}$ ( $1.88 \times 10^{-8}$ )	$6.23 \times 10^{-15}$ ( $6.61 \times 10^{-15}$ )	$7.19 \times 10^{-20}$ ( $5.72 \times 10^{-20}$ )	<b>6.35</b> $\times 10^{-26}$ ( <b>5.39</b> $\times 10^{-26}$ )					



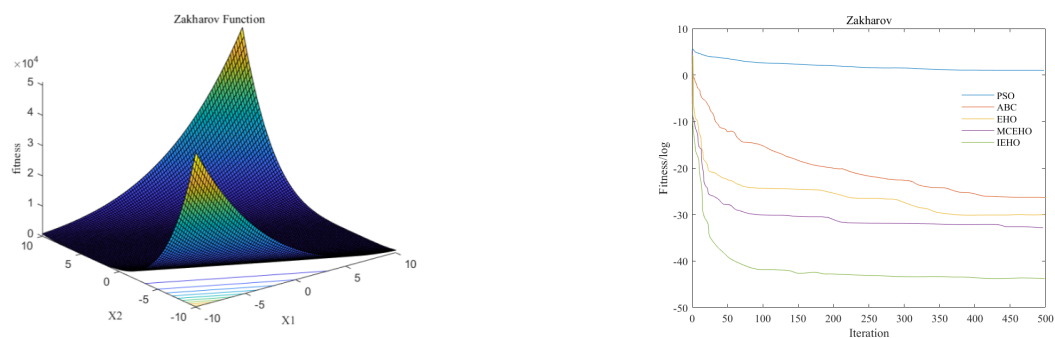
(a) Convergence curve for the  $f_1$  function



(b) Convergence curve for the  $f_2$  function

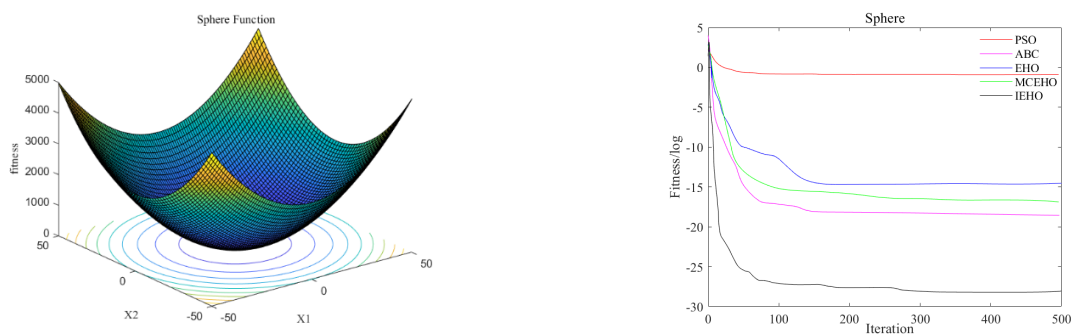


(c) Convergence curve for the  $f_3$  function

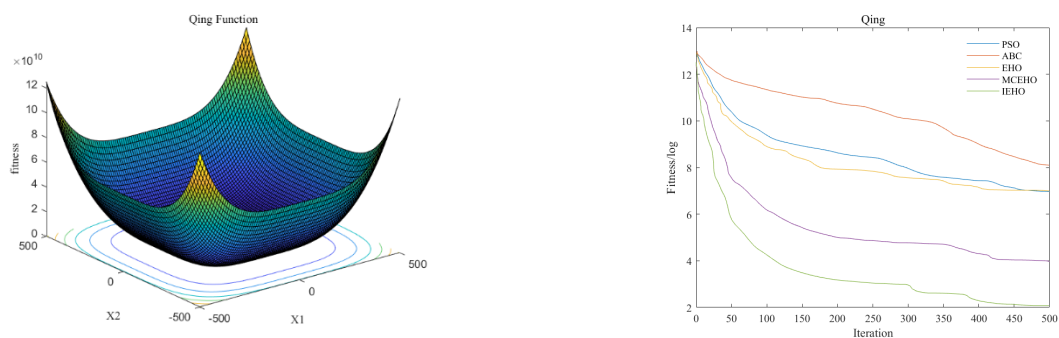


(d) Convergence curve for the  $f_4$  function

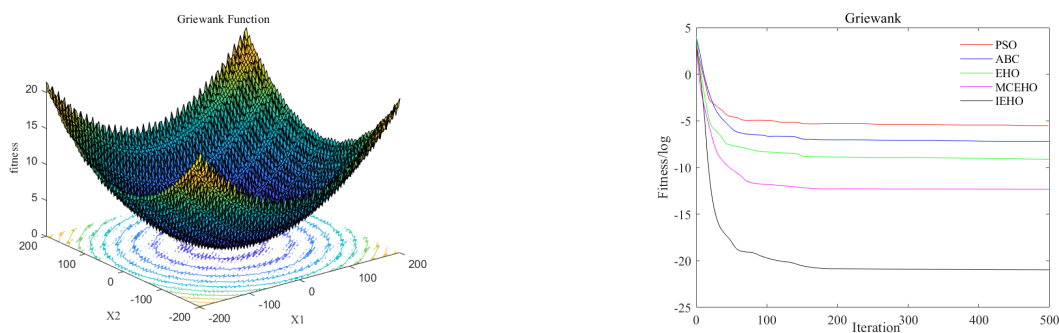
**Figure 4.** Optimization process for the five algorithms when applied to  $f_1 - f_4$  with  $D = 30$ .



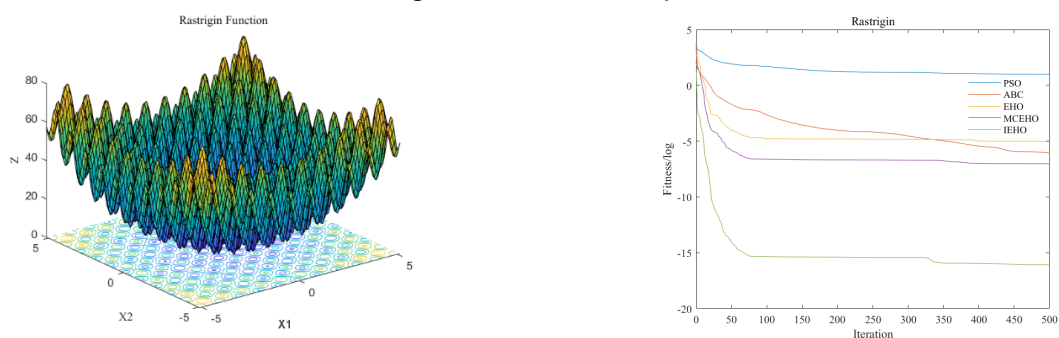
(a) Convergence curve for the  $f_5$  function



(b) Convergence curve for the  $f_6$  function



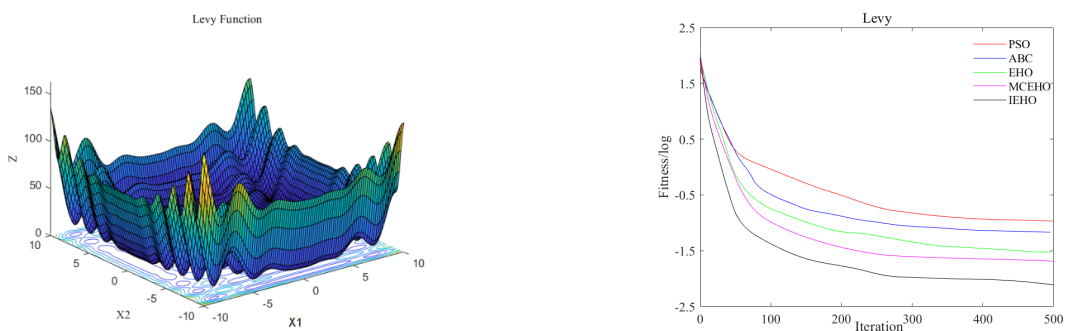
(c) Convergence curve for the  $f_7$  function



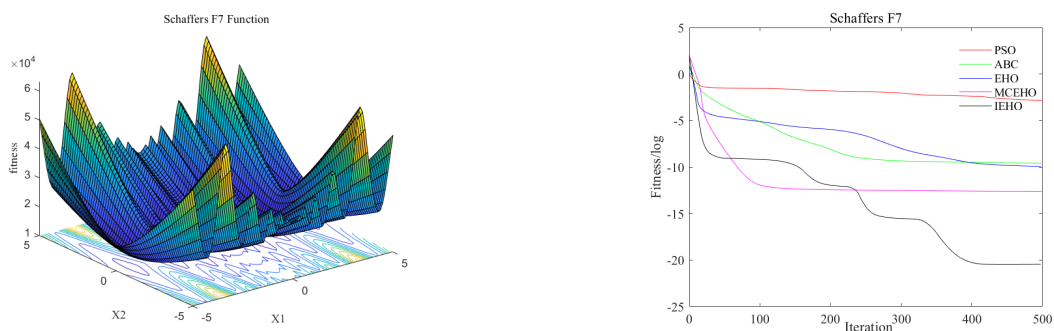
(d) Convergence curve for the  $f_8$  function

**Figure 5.** Optimization process for the five algorithms when applied to  $f_5 - f_8$  with  $D = 30$ .

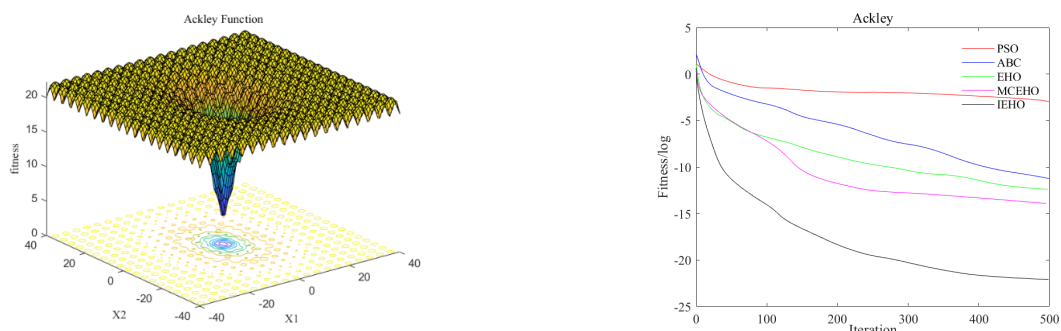




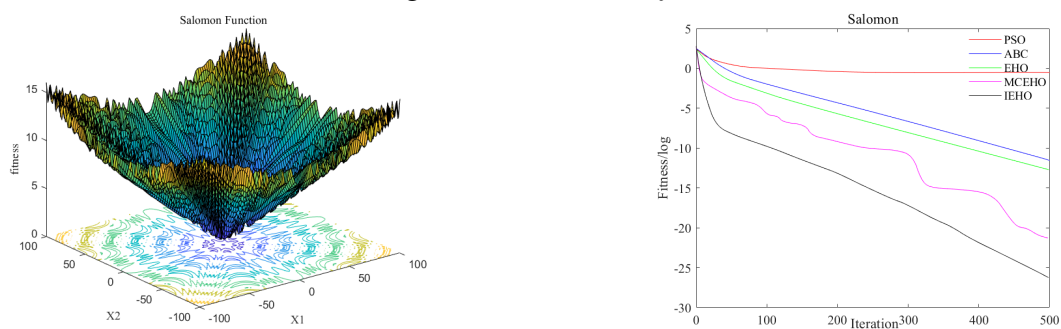
(a) Convergence curve for the  $f_9$  function



(b) Convergence curve for the  $f_{10}$  function



(c) Convergence curve for the  $f_{11}$  function



(d) Convergence curve for the  $f_{12}$  function

**Figure 6.** Optimization process for the five algorithms when applied to  $f_9 - f_{12}$  with  $D = 30$ .

For the multimodal functions  $f_7 - f_{12}$ , the convergence speed and convergence precision of EHO were better than those for PSO, indicating that EHO has obvious advantages over PSO when it comes to solving multi-extremum problems. The EHO and ABC algorithms had similar convergence accuracies, except for the test functions  $f_6$  and  $f_{10}$ . Regarding the other functions, EHO was found to be superior to ABC optimization, as it resulted in a faster convergence speed and higher convergence accuracy. However, EHO is susceptible to the local optima problem; MCEHO can jump out of partial local optima but it is still hard to find global optima. The IEHO algorithm's optimization speed and precision were found to be superior to those of the MCEHO algorithm, indicating that adaptive step grouping learning and an elite retention strategy can effectively improve the algorithms ability to develop global optimal and jump out of local optima.

As can be seen in Table 3, for the unimodal and multi-peak test functions, the IEHO algorithm had the lowest means and standard deviations among all of the algorithms. Its mean value was closest to the theoretical value and able to find the minimum point. As the number of dimensions of the test function increases, the model complexity exponentially grows and increases the difficulties of optimization. To further validate IEHO performance on complex high-dimensional problems, we added a 100D high-dimensional test function experiment; the parameter settings are the same as those for lower dimensions. According to the data in Table 3, the IEHO algorithm performed poorly for  $f_6$  with low convergence accuracy. Regarding the other test functions, it was able to maintain good optimization stability and was found to be superior to the other four algorithms, which fully verified the advantages of IEHO as a tool to solve high-dimensional problems. From the above analysis, it can be seen that the IEHO algorithm presented in this paper has a better convergence effect for both in terms of mean and standard deviation, on low- and high-dimensional test functions, as well as a stronger ability to find the optima as compared with the algorithms based on PSO, ABC, EHO and MCEHO.

## 5. Motor system energy efficiency rating

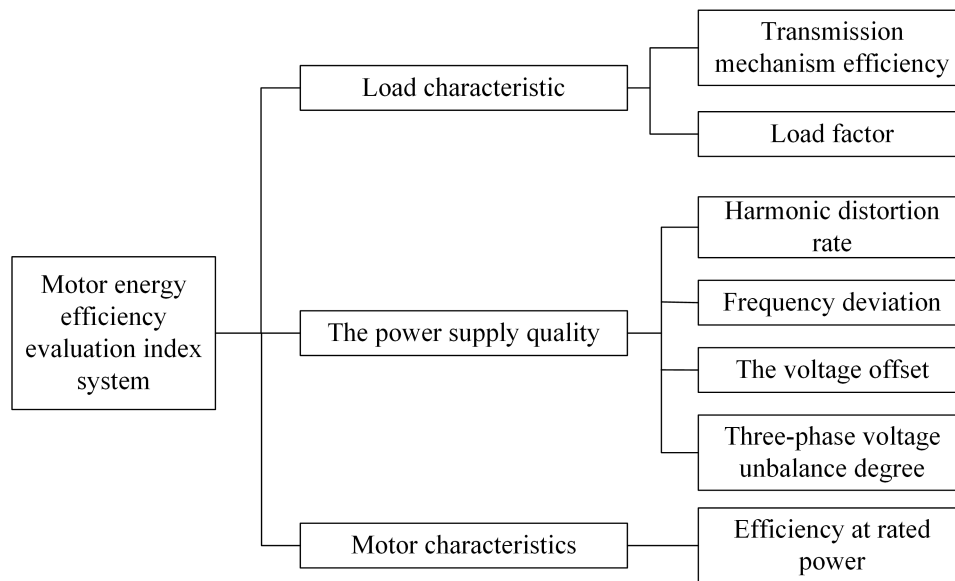
### 5.1. Motor energy efficiency evaluation index system model

In a motor system, the transmission process for the current from the power supply to the motor to the load will generate energy loss, and the energy efficiency of the motor is affected by many factors [32]. In this study, the power quality, motor characteristics and load characteristics of the motor system were considered as the first-level indexes. From these three aspects, seven secondary indexes that have a greater impact on motor energy efficiency were selected to obtain the motor energy efficiency rating system as shown in Figure 7.

The motor energy efficiency has been divided into 10 grades according to the national standards, the upper and lower limits of each indicator's deviation and the impact on the motor energy efficiency. The quantified motor energy efficiency grading standards are based on the parameter values of various indicators at different levels, as shown in Table 4.

The data of the training model were randomly generated. A total of 1000 samples were randomly generated in each indicator interval and marked as  $\{(x_1^i, x_2^i, \dots, x_7^i), y^i\}$ , where  $x_j^i$  represents the random number,  $j = 1, 2, \dots, 7$  represents the seven characteristic index components of the corresponding sample,  $i = 1, 2, \dots, 1000$  represents the corresponding 1000 sample quantity and  $y^i = 1, 2, \dots, 10$  represents the corresponding sample label. First, the training set and test set data were divided and preprocessed; then, the IEHO-SVM model was trained with the training data; finally, the model was verified with the

test data.



**Figure 7.** Motor energy efficiency evaluation system.

**Table 4.** Motor energy efficiency classification standard.

Grade	Load factor	Transmission efficiency	Harmonic distortion rate %	Three-phase voltage unbalance degree %	Voltage offset %	Frequency error %	Rated efficiency %
1	(0.70, 0.80]	0.98	1.00	1.00	2.00	0.10	89.50
2	(0.80, 0.90]	(0.96, 0.98]	(1.00, 2.00]	(1.00, 2.00]	(2.00, 3.00]	(0.10, 0.20]	(88.30, 89.5]
3	(0.60, 0.70]	(0.94, 0.96]	(2.00, 3.00]	(2.00, 3.00]	(3.00, 4.00]	(0.20, 0.30]	(87.1, 88.3]
4	(0.90, 1.00]	(0.92, 0.94]	(3.00, 4.00]	(3.00, 4.00]	(4.00, 5.00]	(0.30, 0.40]	(85.9, 87.1]
5	(0.50, 0.60]	(0.90, 0.92]	(4.00, 5.00]	(4.00, 5.00]	(5.00, 5.50]	(0.40, 0.50]	(84.7, 85.9]
6	(0.45, 0.50]	(0.89, 0.90]	(5.00, 5.50]	(5.00, 5.50]	(5.50, 6.00]	(0.50, 0.55]	(83.5, 84.7]
7	(0.40, 0.45]	(0.88, 0.89]	(5.50, 6.00]	(5.50, 6.00]	(6.00, 6.50]	(0.55, 0.60]	(82.30, 83.50]
8	(0.35, 0.40]	(0.87, 0.88]	(6.00, 6.50]	(6.00, 6.50]	(6.50, 7.00]	(0.60, 0.65]	(81.10, 82.30]
9	(0.30, 0.35]	(0.86, 0.87]	(6.50, 7.00]	(6.50, 7.00]	(7.00, 7.50]	(0.65, 0.70]	(80.00, 81.10]
10	0.30	0.86	7.00	7.00	7.50	0.70	80.00

### 5.2. Improved elephant herding optimization and support vector machine rating model

The SVM is a two-class classification model [33], and it defined as a linear classifier with the maximum interval in the feature space. Different from the perceptron SVM, the learning strategy is to find the separation hyperplane with the maximum interval, which is essential in solving the optimal problem of convex quadratic programming. For one-class classification models, support vector data description is usually used [34, 35].

There is no completely linearly separable data. By introducing a “soft interval”, incomplete linearly separable data can be solved by allowing partial error points. For some linear indivisible problems, it may be nonlinearly divisible by using nonlinear kernel functions to map the original feature space to the higher-dimensional Hilbert space [36]; a nonlinear SVM can be obtained by using a linear SVM, so the nonlinear separable problem is transformed into a linearly separable problem.

Taking a nonlinear SVM as an example, the learning algorithm is as follows:

Input: Training set  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , where  $x_i \in R, y_i \in \{-1, 1\}$ .

Output: Separation hyperplane and classification decision function.

Step 1: Select a proper Gaussian kernel width  $\sigma$  and penalty factor  $c$  and construct and solve convex quadratic programming problems.

$$\begin{aligned} \max_a \quad & \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [a_i y_i \kappa(x_i, x_j) y_j a_j] \\ \text{s.t.} \quad & \sum_{i=1}^n a_i y_i = 0, 0 \leq a_i \leq c \end{aligned} \quad (5.1)$$

Get the optimal solution  $a^* = (a_1^*, a_2^*, \dots, a_n^*)^T$ .

Step 2: Select the  $a$  component of  $a^*$ ,  $a_i^*$  and calculate  $w^*$  and  $b^*$ :

$$\begin{aligned} w^* &= \sum_{i=1}^n a_i^* y_i \kappa(x_i, x_j) \\ b^* &= y_i - \sum_{i=1}^n a_i^* y_i \kappa(x_i, x_j) \\ \kappa(x_i, x_j) &= \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \end{aligned} \quad (5.2)$$

Step 3: Compute the separation hyperplane:

$$w^{*T} \cdot \phi(x) + b^* = 0 \quad (5.3)$$

Step 4: Compute the classification decision function:

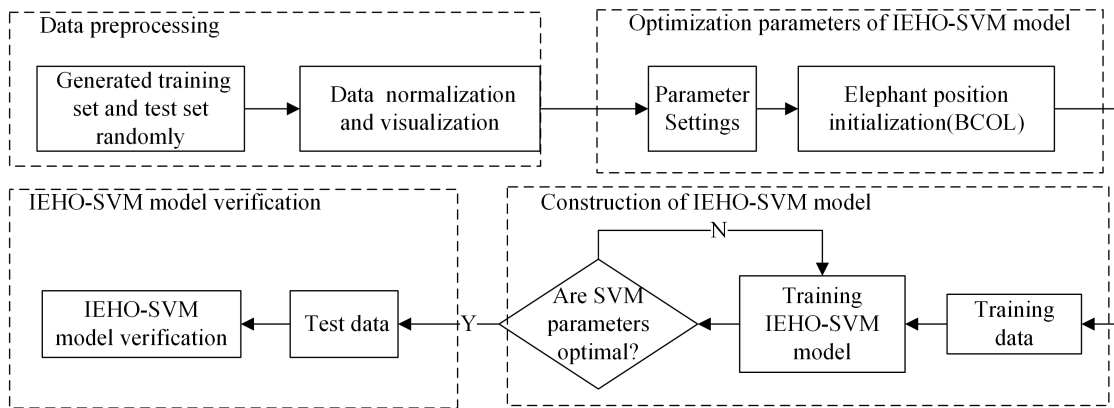
$$f(x) = \text{sign}(w^{*T} \phi(x) + b^*) \quad (5.4)$$

According to the above analysis, the main parameters affecting SVM learning ability are the penalty factor  $c$  and Gaussian kernel width  $\sigma$ . The selection of the parameters has an important influence on the running result of the SVM model, which determines its learning ability and generalization ability. The penalty factor  $c$  represents the tolerance of error, where a larger  $c$  means a higher likelihood of overfitting, and a smaller  $c$  means a higher likelihood of underfitting. The Gaussian kernel width  $\sigma$ , which is inversely proportional to the number of support vectors, affects the model training and prediction speed. Traditional parameter selection methods include manual selection and grid optimization, which are more suitable for small data. When the amount of data increases, the amount of computation and calculation time will also increase and it becomes easier to fall into local optimization.

In this study, the IEHO algorithm was used to optimize SVM parameters  $c$  and  $\sigma$ , and the algorithm was applied to the motor energy efficiency rating. The specific algorithm process is shown in Figure 7. The position vector for each elephant corresponds to a group of specific parameter combinations. The evaluation function for its location fitness is defined as follows:

$$F = \frac{1}{N} \sum_{i=1}^N \|Y_i - Y_i^*\|^2 \quad (5.5)$$

where  $N$  represents the number of samples;  $Y_i$  and  $Y_i^*$  represent the actual value and the predicted value of SVM, respectively.



**Figure 8.** Flowchart of the IEHO-SVM model.

### 5.3. Example simulation and result analysis

In order to evaluate the performance of the algorithm and its robustness on the training data, the following two strategies were performed.

1) A stratified sampling method was used to divide the data set into four different proportions: (80%, 20%), (70%, 30%), (60%, 40%) and (50%, 50%).

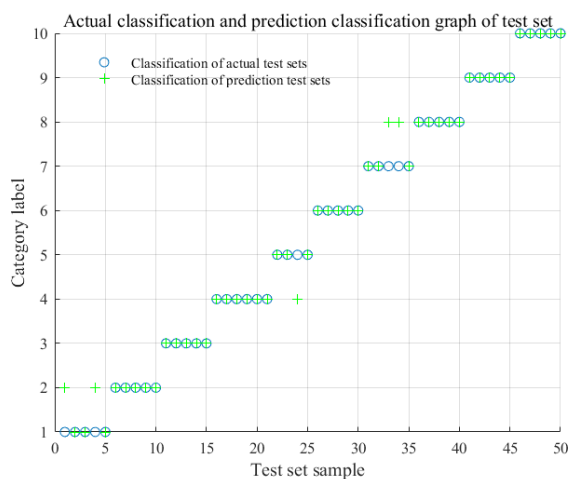
2) Using the same data, the IEHO-SVM algorithm proposed in this paper was compared with the original SVM and k-nearest neighbor [37] algorithms.

The classification accuracies of the three algorithms on different training sets and test sets are shown in Table 5. The classification results for the three algorithms on the 50% training sets and test sets are shown in Figure 8.

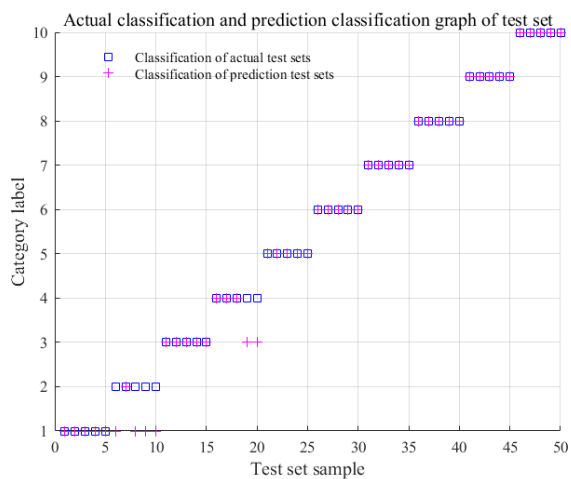
As we can see in Table 5 and Figure 9, the accuracy of the IEHO-SVM algorithm on different training sets and test sets was higher than 98%, whereas that of the KNN algorithm was around 91% and that of the original SVM algorithm was around 87%. This experiment demonstrates that the robustness of the IEHO-SVM algorithm on training data is not limited to a particular training-test set combination, but that better results can be achieved in all these cases. So, we found that the IEHO-SVM algorithm can accurately grade the motor energy efficiency with high accuracy and reliability.

**Table 5.** Accuracies of three algorithms.

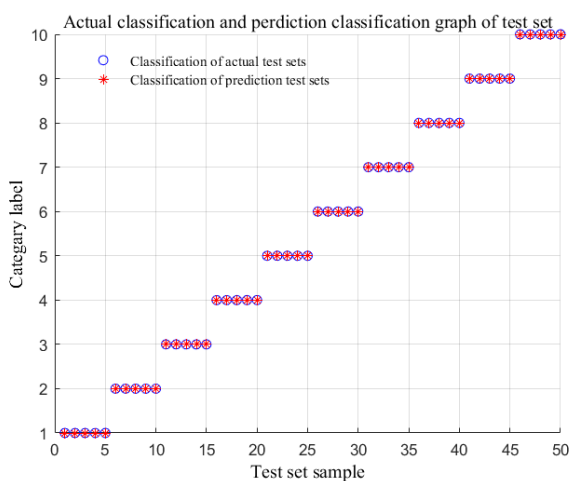
Set of instances	KNN	SVM	IEHO-SVM
80% Train, 20% Test	90%	87%	98%
70% Train, 30% Test	91%	86%	99%
60% Train, 40% Test	90%	88%	100%
50% Train, 50% Test	92%	89%	100%



(a) KNN results



(b) SVM results



(c) IEHO-SVM results

**Figure 9.** Grading results for three algorithms on the 50% training set and test set.

## 6. Conclusions and future prospects

In the past, research on motor energy efficiency mainly focused on the rated operating condition, but there were few studies on motor energy efficiency performed under off-conditions. The correct grade of the motor energy efficiency can provide scientific guidance for motor energy-saving upgrades, fault damage and the elimination of backward motors. To determine an accurate grade of the motor energy efficiency, we selected seven evaluation indexes that have a great influence on motor energy efficiency from the three aspects of power quality, load characteristics, and motor characteristics, and established a quantitative classification standard.

Aiming to eliminate disadvantages of the original EHO algorithm, such as a slow convergence speed, a low convergence accuracy and susceptibility to the local optima problem, and developed five improvement measures. A new EHO algorithm, i.e., the IEHO algorithm, was designed and compared with other algorithms. The results show that the IEHO algorithm has a higher convergence precision and faster convergence speed than other algorithms, can jump out of local optima and exhibits better performance for optimization.

The IEHO algorithm was used to optimize the SVM parameters, and the IEHO-SVM model was established; it was subsequently applied to the motor energy efficiency rating system. Compared with the KNN and original SVM algorithms, the experimental results show that the IEHO-SVM algorithm can accurately grade the motor energy efficiency with an accuracy higher than 98%, which was better than the KNN and original SVM algorithms.

The IEHO algorithm and IEHO-SVM model proposed in this paper has strong portability, and it can not only be used for motor energy efficiency rating, but also for other optimization problems. The motor energy efficiency grading model established in this paper is simple and easy to realize, has a high practical value and is of great significance for realizing energy conservation and emission reduction and improving energy utilization.

Possible opportunities and future proposals are as follows. Initialization strategies for orthogonal learning can be attempted next, and location updates can be attempted for quantum computing, parallel computing, and co-evolutionary approaches. The IEHO algorithm presented in this paper been demonstrated to effectively solve static optimization problems, and the development of dynamic optimization and variants of multi-objective optimization methods that can handle more kinds of optimization problems can be attempted next.

### Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

1. X. Cui, Y. Luo, Y. Yang, Y. Guo, H. Wang, X. Liu, Energy saving mechanism and energy saving approach of asynchronous motor under periodic variable working conditions, *Proc. Chin. J. Electr. Eng.*, **28** (2008), 90–97.

2. D. Yang, H. R. Karimi, L. Gelman, A fuzzy fusion rotating machinery fault diagnosis framework based on the enhancement deep convolutional neural networks, *Sensors*, **22** (2022), 671. <https://doi.org/10.3390/s22020671>
3. Z. Lv, Online monitoring of batch processes combining subspace design of latent variables with support vector data description, *Complex Eng. Syst.*, **1** (2021). <https://org.doi/10.20517/ces.2021.02>
4. J. Yu, X. Yan, Data-feature-driven nonlinear process monitoring based on joint deep learning models with dual-scale, *Inf. Sci.*, **591** (2022), 381–399. <https://doi.org/10.1016/j.ins.2021.12.106>
5. J. Yu, X. Yan, Active features extracted by deep belief network for process monitoring, *ISA Trans.*, **84** (2018), 247–261. <https://doi.org/10.1016/j.isatra.2018.10.011>
6. J. Yu, X. Yan, Multiscale intelligent fault detection system based on agglomerative hierarchical clustering using stacked denoising autoencoder with temporal information, *Appl. Soft Comput.*, **95** (2020), 106525. <https://doi.org/10.1016/j.asoc.2020.106525>
7. Y. Zhao, X. Li, S. Yang, Minimum allowable values of energy efficiency and energy efficiency grades for small and medium three-phase asynchronous motors 18613, 2012.
8. C. Luo, W. B. Ma, J. Zhao, Evaluation and analysis of the influence factors on the energy consumption of the motor system, *Motor Control Appl.*, **43** (2016), 98–103.
9. C. Li, Research on energy saving evaluation index system of motor system, *Motor Control Appl.*, **43** (2016), 74–77.
10. L. X. Ma, M. Y. Lv, Research on intelligent quantification and classification method of power energy efficiency, *Power Sci. Eng.*, **33** (2017), 46–49.
11. J. Cervantes, F. Garcia-Lamont, L. Rodriguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, *Neurocomputing*, **408** (2020), 189–215. <https://doi.org/10.1016/j.neucom.2019.10.118>
12. G. G. Wang, S. Deb, Z. Cui, Monarch butterfly optimizations, *Neural Comput. Appl.*, **31** (2019), 1995–2014. <https://doi.org/10.1007/s00521-015-1923-y>
13. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
14. S. H. S. Moosavi, V. K. Bardsiri, Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation, *Eng. Appl. Artif. Intell.*, **60** (2017), 1–15. <https://doi.org/10.1016/j.engappai.2017.01.006>
15. G. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Comput.*, **10** (2018), 151–164. <https://doi.org/10.1007/s12293-016-0212-3>
16. F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. AlAtabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, *Future Gener. Comput. Syst.*, **101** (2019), 646–667. <https://doi.org/10.1016/j.future.2019.07.015>
17. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. <https://doi.org/10.1016/j.future.2019.02.028>



18. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. <https://doi.org/10.1016/j.future.2020.03.055>
19. I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.*, **181** (2021), 115079. <https://doi.org/10.1016/j.eswa.2021.115079>
20. J. Tu, H. Chen, M. Wang, A. H. Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **18** (2021), 674–710. <https://doi.org/10.1007/s42235-021-0050-y>
21. I. Ahmadianfar, A. AsgharHeidari, S. Noshadian, H. Chen, A. HGandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. <https://doi.org/10.1016/j.eswa.2022.116516>
22. G. G. Wang, S. Deb, X. Z. Gao, L. Coelho, A new metaheuristic optimization algorithm motivated by elephant herding behavior, *Bio-Inspir. Comput.*, **8** (2016), 394–409. <https://doi.org/10.1504/IJBIC.2016.081335>
23. Z. Zhang, H. Wang, H. Zhou, S. You, Parameter estimation of chaotic systems based on multi-mechanism hybrid image group Algorithm, *Microelectron. Comput.*, **6** (2020), 40–45.
24. M. S. Tavazoei, M. Haeri, Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms, *Appl. Math. Comput.*, **187** (2007), 1076–1085. <https://doi.org/10.1016/j.amc.2006.09.087>
25. F. Chakraborty, P. K. Roy, D. Nandi, Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding, *Evol. Intell.*, **12** (2019), 445–467. <https://doi.org/10.1007/s12065-019-00238-1>
26. W. Luo, H. Jin, H. Li, R. Zhou, Blind source separation of radar signals based on chaotic adaptive firework algorithm, *Syst. Eng. Electr.*, **42** (2020), 2497–2505.
27. F. Marini, B. Walczak, Particle swarm optimization (PSO). A tutorial, *Chemom. Intell. Lab. Syst.*, **149** (2015), 153–165. <https://doi.org/10.1016/j.chemolab.2015.08.020>
28. J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective RealParameter Numerical Optimization, in *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University*, (2014).
29. N. Lynn, P. N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, *Swarm Evol. Comput.*, **24** (2015), 11–24. <https://doi.org/10.1016/j.swevo.2015.05.002>
30. Y. Xue, J. Jiang, B. Zhao, T. Ma, A self-adaptive artificial bee colony algorithm based on global best for global optimization, *Soft Comput.*, **22** (2018), 2935–2952. <https://doi.org/10.1007/s00500-017-2547-1>
31. W. F. Gao, L. L. Huang, S. Y. Liu, C. Dai, Artificial bee colony algorithm based on information learning, *IEEE Trans. Cybern.*, **45** (2015), 2827–2839. <https://doi.org/10.1109/TCYB.2014.2387067>

32. Y. Luo, H. Jin, H. Li, H. Rong, Blind source separation of radar signals based on chaotic adaptive fireworks algorithm, *Syst. Eng. Electron. Technol.*, **42** (2020), 95–103.
33. J. Peng, Y. Zhou, C. L. P. Chen, Region-kernel-based support vector machines for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sensing*, **53** (2015), 4810–4824. <https://doi.org/10.1109/TGRS.2015.2410991>
34. Z. Lv, X. Yan, Q. Jiang, Batch process monitoring based on self-adaptive subspace support vector data description, *Chemom. Intell. Lab. Syst.*, **170** (2017), 25–31. <https://doi.org/10.1016/j.chemolab.2017.09.009>
35. J. Yu, X. Yan, Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network, *IEEE Trans. Cybern.*, **50** (2020), 3998–4007. <https://doi.org/10.1109/TCYB.2019.2948202>
36. Y. Shao, C. Zhang, X. Wang, N. Deng, Improvements on twin support vector machines, *IEEE Trans. Neural Networks*, **22** (2020), 962–968. <https://doi.org/10.1109/TNN.2011.2130540>
37. S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient kNN classification with different numbers of nearest neighbors, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2018), 1774–1785. <https://doi.org/10.1109/TNNLS.2017.2673241>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)